Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 4, August 2026, pp. 3719~3731
ISSN: 2302-9285, DOI: 10.11591/eei.v15i4.11767

Evaluating lexical feature extraction for plagiarism detection in Arabic documents

Marwah Alian[1,2], Dana Halabi[1], Hadeel Rida Alshboul[1]
[1]Department of Computer Science, Faculty of Information Technology, The World Islamic Sciences and Education University, Amman, Jordan
[2]Department of Basic Sciences, Faculty of Science, The Hashemite University, Zarqa, Jordan

Article Info

Article history:

Received Oct 31, 2025
Revised Jun 14, 2026
Accepted Jul 23, 2026

Keywords:

Arabic documents
Artificial neural network
Longest common subsequence
Plagiarism detection
Support vector machine

ABSTRACT

Plagiarism detection is the task of determining whether a document contains parts from other documents by employing different styles of plagiarism, such as copying certain parts and reordering or replacing words with synonyms, without citing the original text owner. This task is important in many applications, and there are two primary types of plagiarism detection methods: external and intrinsic. Plagiarism detection in Arabic documents is challenging because of Arabic's rich morphological features, lexical variation, and syntactic complexity, which limit the effectiveness of some detection approaches. To address these challenges, this study introduces an external plagiarism detection framework built on an artificial neural network (ANN) model and a lexical feature extraction framework adapted to the linguistic features of Arabic. The proposed framework is evaluated using ExAraPlagDet-2015 benchmark, where a baseline model using support vector machine (SVM) is introduced for comparison. Experimental results demonstrate notable improvements in plagiarism detection performance of the proposed framework compared with SVM and other baseline methods. The proposed framework provides a precision value of 92% and an F-score value of 96%, verifying its effectiveness for Arabic plagiarism detection.

Corresponding Author:

Marwah Alian
Department of Computer Science, Faculty of Information Technology
The World Islamic Sciences and Education University
Amman, Jordan
Email: Marwah.alian@wise.edu.jo

1. INTRODUCTION

Academic and professional contexts depend heavily on plagiarism detection, which plays a vital role in maintaining academic integrity and ethical standards [1], [2]. Unauthorized text reuse has become more common as a result of the rapid advancement of digital Arabic content in academic institutions and websites. Therefore, detecting how much information is shared between two texts is a key component of many natural language processing applications such as semantic similarity, sentiment analysis [3], question answering [4], plagiarism detection [5], and others.

Plagiarism happens when someone takes the writing of someone else and claims it as their own [6]. Several forms of plagiarism observed in documents, including text copy-paste, paraphrasing via synonym substitution, use of fake references, translation-based plagiarism, and idea plagiarism [7].

Plagiarism detection systems are used to check what the plagiarist is attempting to hide using multiple forms of plagiarism, such as rewording, synonyms substitution, and paraphrasing [8]. In the academic setting,

plagiarism and research integrity are sensitive topics, particularly with the rapid advancement of large language models (LLMs) and artificial intelligence (AI) [9].

The two main techniques for plagiarism detection are external and intrinsic. In the context of external plagiarism detection, the aim is to detect matches among the suspicious document one or more source document, whereas intrinsic plagiarism detection focuses on detecting variations in writing style between the suspect and source documents. The majority of research on Arabic plagiarism detection focuses on external plagiarism [10]. For external plagiarism detection, much research represents a document as a vector with n dimensions and then measures similarity between vectors. Intrinsic plagiarism detection is used to detect multilingual plagiarism cases [11].

Plagiarism detection in natural languages is more difficult than plagiarism detection in programming languages because of the morphology and flexibility in human language. Detecting plagiarism in Arabic texts is important because Arabic is widely used in domains such as education and culture. Also, available plagiarism detection systems work for English text but fail to capture the morphological and lexical features in Arabic text. Detecting plagiarism in Arabic is more difficult than in English because of language complexity, morphological features, and diverse writing styles [12].

Plagiarism detection is important to maintain originality and reinforcing the reliability of academic and professional work, but effective detection systems specifically customized to Arabic texts are still limited. This research proposes a framework to fill the gap in external Arabic plagiarism detection by combining simple approaches including n-gram overlap and longest common subsequence (LCS), which are widely used in lexical similarity metrices with an artificial neural network (ANN) model. Arabic universities face increasing plagiarism cases, yet existing tools are English-centric. The proposed framework tries to fill this gap.

The proposed work benefits from neural networks' capability to manage noisy or incomplete data, which enhances robustness against specific weaknesses. It is evaluated using ExAraPlagDet-2015 benchmark, which was released by the shared task "AraPlagDet" mainly to evaluate external and intrinsic plagiarism detection methods [10]. The results of the proposed work show an outperformance over existing approaches applied to ExAraPlagDet-2015 benchmark.

Unlike previous work that relied on term frequency-inverse document frequency (TF-IDF) or word embeddings, the proposed method combines lexical features and n-gram overlap with ANN learning to capture similarities at both shallow and deep. This improves robustness against lexical substitution and paraphrasing, two techniques frequently employed in Arabic plagiarism.

The structure of this research is structured as follows: section 2 review and compares related work relevant to the introduced study of plagiarism detection, section 3 describes the research methodology, and the proposed method and measures used, section 4 illustrates and discusses the proposed work while section 5 describes the experimental setup and results. Finally, the paper concluded in section 6.

2. RELATED WORK

Particularly in the context of the present driven by recent progress in AI technology and LLMs, plagiarism and research integrity are challenging subjects in educational environments [9]. However, the majority of work in detecting plagiarism in documents is performed for academic aspects specially to judge postgraduates' work.

Bin-Habtoor and Zaher [8] discussed several issues in plagiarism such as plagiarism in documents, source code plagiarism detection algorithms. They categorize the approaches for the assessment of source code similarity into two types; the first compare different programming languages' source codes, while the other type handles modifications in codes, but a longer time is required. The plagiarism techniques are based on techniques that detect the similarity, such as attribute counting techniques that create special "fingerprints" for collection files. Many algorithms for detecting plagiarism in textual documents are based on string comparisons, in contrast, method for detecting source code plagiarism rely on software metrics, then extracting a set of features to detect plagiarism in programs.

Abdelrahman *et al.* [1] proposed an approach to detect plagiarism in Arabic documents, which is considered a content-based approach. In this framework, word stemming, fingerprint, and heuristic algorithms are used to analyze Arabic documents across multiple linguistic levels: sentence, paragraph, and document. The main components of the proposed framework are preprocessing, fingerprinting, document representation, and comparison of similar terms. The preprocessing stage consists of stop word removal, tokenization, stemming, and synonym replacement using Arabic wordnet (AWN) for hidden plagiarism. Also, some hidden plagiarism, such as the change in the structure of the sentence and replacement of synonyms is treated.

Al-Thwaib *et al.* [2] constructed a plagiarism corpus called JUPlag to be used in the assessment of Arabic plagiarism detection techniques. The constructed corpus consists of 2312 dissertations collected from the institutional repository of the University of Jordan (JU). JUPlag corpus uses the adopted Dewey decimal

classification system to classify its content, which is similar to that of JU Library. In JUPlag, the dissertations are categorized by subject area, allowing plagiarism detection to be done on a sub-corpus rather than the complete corpus. The final output dataset consists of segmentation of n-gram, annotation of metadata, followed by the morphological analysis and annotation of each word in the collected dissertations.

Hattab [13] proposed a system based on latent semantic indexing (LSI) to detect cross-language plagiarism. A cross-language semantic space is built, then the semantic similarity between English and Arabic documents. Documents are presented in the semantic space as vectors using LSI, with identical vectors having a cosine similarity value of one. English and Arabic documents (1000) are used in the experiment to evaluate the presented approach. The obtained results indicate that for cross-language English-Arabic plagiarism detection, the proposed LSI-based model achieves 93% similarity.

Siddiqui *et al*. [14] constructed an Arabic corpus for plagiarism detection. The corpus includes 1665 Arabic documents collected from students' assignments in King Abdulaziz City of Science and Technology (KACST), which provides several types of plagiarism. These documents are classified into source and suspicious documents. The corpus has various form of plagiarism, such as direct copying, small, and heavy modifications.

Yousef and Aziz [15] proposed a TF-IDF based framework for Arabic plagiarism detection. This framework involves preprocessing, synonym replacement, term weighting, and similarity measurement. The term weighting procedure relies on the TF-IDF approach and then cosine similarity measure between the source and suspicious documents representing the two documents as m-dimensional vectors over the terms set. They evaluate their proposed framework using the contemporary Arabic (CCA) corpus with 200 documents selected from several topics. The TF-IDF plagiarism detection system with a synonym dictionary achieves an accuracy of 92%.

Ghanem *et al*. [16] proposed a Arabic plagiarism detection system based on hybrid approach, referred to as hybrid plagiarism detection system (HYPLAG) that combines knowledge-based with corpus-based information to detect Arabic plagiarism. HYPLAG consists of a number of phases that are pre-preprocessing, indexing, stemming, part-of-speech (POS) tagging, and sentence ranking. The highest rank is achieved by the sentence with the highest relevance to the terms where cosine similarity is measured using two vectors, one for nouns and the other for verbs and TF-IDF term weighting. ExAraPlagDet-2015 dataset is used to evaluate the proposed system which attains a recall of 87%, a precision value of 92%, and an F-score value of 89%.

Nagoudi *et al*. [5] proposed framework detects plagiarism in Arabic text through two successive layers: fingerprinting and word embedding. The first stage produces fingerprint for each sentence, starting by n-grams chunking, then adding only the least frequent n-grams are selected to reduce the inverted index size, then applying a hashing function is applied to the selected n-grams to generate the fingerprint. The second layer uses word embedding with words alignment and words weighting using IDF and POS weighting methods to measure the similarity between sentences. The proposed approach attains a recall value of 88% and a precision value of 85%.

Alzahrani and Salim [17] developed an approach for detecting Arabic texts plagiarism. This approach has four steps: preprocessing, suspicious document retrieval, and similarity measurement. The similarity between suspicious and source documents is based on comparing their sentences while the similarity between sentences is based on a fuzzy degree, which is defined in the range of [0,1]. Two sentences are considered as plagiarized when their similarity score exceeds that passes a predefined threshold. To evaluate their approach, they perform their experiment using the PAN'09 training corpus, and the results show a recall value of 31% and a precision value of 54%.

Khan *et al*. [11] proposed a plagiarism detection system developed for Arabic text. This system consists of a global part that is heuristics-based and a local part where the similarity between the source documents and suspicious documents is measured. At the global level, an information retrieval system is employed to retrieve source documents after generating a set of queries using Google search API. In the local part, the retrieved source documents are downloaded, and a detailed similarity report is produced where plagiarized texts from a suspicious document are highlighted with several colors to indicate source documents.

Suleiman *et al*. [18] presented a plagiarism detection approach relying on using word2vec to encode words into vectors, then the words similarity is measured as the cosine similarity to detect plagiarism. To evaluate their approach, they conduct their experiment using OSAC corpus to train the word2vec model and generate words' vectors. They select documents from the OSAC corpus, and they try to generate suspicious documents by altering the order of words or replacing some words by others with identical or different meanings, or by replacing singular with plural. The results for the tested documents show a precision value of 99%.

Omoush *et al*. [19] the research study examines the effectiveness of several machine learning and statistical techniques, such as word embedding, TF-IDF, bigram, and unigram. The bigram approach was the most effective, according to experiments. Furthermore, on every metric, every machine learning classifier performed outperforms the bigram method.

Rashidy *et al*. [20] proposed a novel plagiarism detection approach to identify the most accurate similarity cases by extracting the selection of the most relevant similarity features and the construction of a hyperplane equation based on the selected features. A number of benchmarks were used to assess the presented system. The findings confirm that the proposed approach obtained the best Plagdet with a scores of 89.12%, 92.91% and F-measure values of 89.34% and 92.95% over PAN 2013 and PAN 2014 datasets, respectively.

To respond to the queries in these datasets, they employed the ChatGPT 3.5 model during its initial release phase. But when the AI-generated text (AIGT) section of the English-language human ChatGPT comparison corpus (HC3) datasets was examined, some entries were found to be absent or incomplete ChatGPT-generated information, yielding only 20,982 samples rather than the 26,903 they stated in [21]. Table 1 provides a comparative summary of previous research in terms of the detection approach, target language, dataset, evaluation metrics, and reported results.

Table 1. Comparison of previous research in terms of approach for detecting plagiarism

Ref.	Approach	Language	Dataset	Evaluation metrics	Results (%)
[1]	Content-based approach: fingerprint and heuristic algorithms	Arabic	Large set of Arabic documents	N/A	N/A
[13]	LSI	Cross-language: English and Arabic	1000 English and Arabic documents	Accuracy	93
[15]	TF-IDF based framework	Arabic	CCA corpus	Accuracy	92
[16]	HYPLAG	Arabic	ExAraPlagDet-2015	Recall Precision F-score	87 92 89
[5]	Two layers system: 1st layer: fingerprint and 2nd layer: words embedding and words alignment	Arabic	ExAraPlagDet-2015	Recall Precision	88 85
[17]	Fuzzy similarity-based approach	Arabic	PAN'09 training corpus	Recall Precision	31 54
[11]	Web based system with a global heuristics-based part and a local part	Arabic	N/A	N/A	N/A
[18]	Word2vec used to represent words as vectors	Arabic	OSAC corpus for training and selected documents for testing	Precision	99
[19]	Word embedding, TF-IDF, bigram, and unigram	Arabic			
[20]	SVM and Chi-square techniques	English	PAN 2013 and PAN 2014	F-score	89.34 and 92.95
[21]	AraELECTRA and XLM-R	Arabic	Large dataset and custom dataset	Accuracy on large dataset	94.9
Ours	N-gram overlap and LCS features combined with ANNs	Arabic	ExAraPlagDet-2015 dataset	Precision Recall F1-score	92 100 96

This research integrates previously used n-gram overlap and LCS, as document-level features within a supervised ANN framework to predict whether two Arabic texts are plagiarized and the proposed approach is evaluated using the ExAraPlagDet-2015 dataset.

3. METHOD

This section describes the proposed framework for detecting plagiarism in Arabic texts by leveraging a combination of lexical features and ANNs. By formulating plagiarism detection as a binary classification challenge, the methodology follows a structured pipeline: beginning with formal task definition, followed by rigorous text preprocessing, hierarchical feature extraction, and finally, the selection of optimal features to feed into the detection models.

3.1. Problem formalization

The primary aim is to identify the presence of plagiarism in Arabic documents by handling this task as a binary classification challenge. We define the detection environment as:

– Document space: let d_{susp} be a suspicious document; D={d_1; d_2; …; d_n} be a set of potential source documents for plagiarized documents.

- The mapping task: the detection model seeks to optimize a similarity function to identify pairs (p, p') where p ϵ d_{susp}, p' ϵ d_{src} such that d_{src} ϵ D.
- Labeling: each pair is assigned a label L ϵ {0, 1}, where 1 denotes "plagiarized" (positive) and 0 denotes "non-plagiarized" (negative).
- Levels of complexity: the model must differentiate between various levels of similarity, ranging from identical copy-pasting (no obfuscation) to complex manual obfuscation involving synonym substitution and paraphrasing.

3.2. Dataset

The proposed approach is evaluated by using the ExAraPlagDet-2015 dataset [10], which is a publicly available benchmark for Arabic plagiarism detection. It was firstly introduced in the shared task of 2015 for the first competition of Arabic plagiarism detection (AraPlagDet).

The dataset covers three types of plagiarism; artificial obfuscation, simulated obfuscation, and no-obfuscation, besides to samples for non-plagiarism. The dataset consists of multiple text files: suspicious documents, source documents, and plagiarism-annotation files. The plagiarism-annotation files contain details about the plagiarized texts and plagiarism types. The ExAraPlagDet-2015 dataset consists of two parts: a training part and a testing part where training part has 567 source documents and 607 suspicious documents, but, the testing part has 570 source documents and 601 suspicious documents.

Two new primer datasets were constructed from ExAraPlagDet-2015 dataset: the primer training dataset and primer testing dataset. These new datasets contain the desired data from suspicious documents, source documents, and plagiarism-annotation files included in the training and testing parts of ExAraPlagDet-2015.

The proposed approach aims to develop a binary classifier for plagiarism selection which means that it is a comparison task between two texts and then to predict whether the first text is plagiarized from the second text or not. Therefore, each primer dataset contains two text columns, and one integer column that is used as a class feature. The texts of the first column are extracted from suspicious documents while the texts of the second column are extracted from source documents. The label value of each pair is extracted from plagiarism-annotation files. It is given a value of one if the suspicious document is similar to the source document and is classified as plagiarized and zero value if the suspicious document does not exhibit evidence of plagiarism. Table 2 represents the class distribution in the primer training and primer testing datasets.

Table 2. Class-wise distribution in the primary training and testing datasets

Class	Training dataset	Testing dataset
No-plagiarism	169	169
Plagiarism	1725	1727

3.3. Preparing training and testing sets

The premier datasets undergo a sequence of phases to prepare them for use in the proposed model. The sequence begins with Arabic text preprocessing, followed by lexical feature engineering, feature extraction, and finally feature selection and correlation analysis. Figure 1 illustrates these phases and their sequence. More details about each phase are discussed in the following subsections.

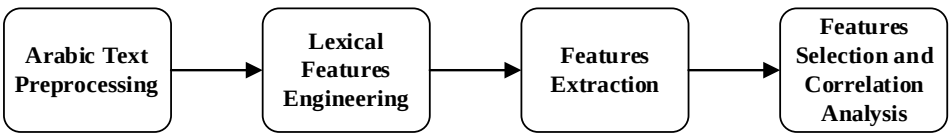


Figure 1. Sequence of phases to prepare datasets

3.3.1. Arabic text preprocessing

In this phase, the text in the primer datasets is cleaned and preprocessed to be prepared for the lexical feature engineering phase. The presented approach is relying on identifying phrases that match in the suspicious document (d_{susp}) and source document (d_{src}), with each document represented as a sequence of words rather than complete sentences. Therefore, a sequence of preprocessing operations is applied to Arabic texts,

including removing stop words, special characters, and diacritics (Fatha (َ), Damma (ُ), Kasra (ِ), Sokoun (ْ), and Shadda (ّ)), normalization, tokenization, and stemming. Figure 2 illustrates these operations.

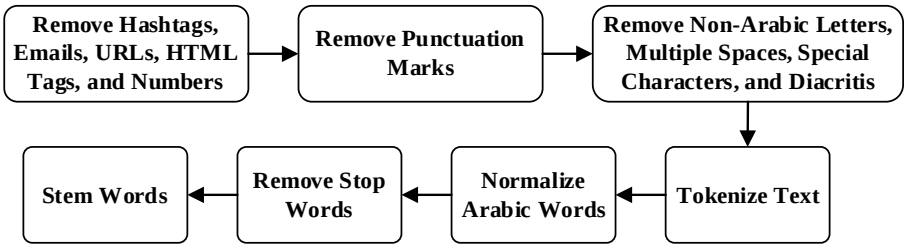


Figure 2. Sequence of preprocessing operations applied to Arabic texts

During normalization, the forms of Arabic letters, such as Aleif (ألف) and Haa (الهاء), that appear in many shapes depending on their position in a word, are converted into a standard shape. Specifically, the shapes of the letter Haa (الهاء) at the end of a word (ـه ,ه) are converted into (ـة ,ة), and Aleif shapes (أ, إ, ا, آ) are converted into (ا).

Table 3 provides an example of a text before and after preprocessing is performed, where the stem [22] of words such as "analysis تحليل" and "economic اقتصاد" appears in the processed text as "analyze حلل" and "intent قصد," respectively.

Table 3. Example of Arabic text before and after preprocessing

Arabic text before preprocessing	Arabic text after preprocessing
يعتمد الاقتصاد كمادة أكاديمية بشكل أساسي على الأساليب الرياضية، إلى جانب اعتماده على الأساليب الأدبية. يتم اعتماد الأساليب الرياضية والكمية لأغراض تحليل اقتصاد ما بدقة، أو لتحليل مناطق بعينها داخل الاقتصاد. وكأمثلة على هذه النماذج والأساليب في التحليل نذكر	الب اقتصاديه الب ريض اقتصاديه كمه عدل عمد قصد كمد اكاديميه شكل سسي علي الب ريض الي جنب عمد علي الب ادب يتم عمد الب ريض كمه غرض حلل قصد بدق حلل نطق بعن دخل قصد كامثل علي اذج الب حلل ذكر

3.3.2. Lexical feature engineering

Lexical features involved in the plagiarism detection model include containment features and the LCS feature [23].

a. Containment features

An n-gram is a sequential grouping of words. We define a containment of n-gram for a suspicious document (d_{susp}) and a source document (d_{src}) as the total number of n-grams that appear in both documents. This containment measures the overlap between the two documents and is calculated as in (1) [23]:

$$C_{\_n}(d_{susp}, d_{src}) = \frac{|S(d_{susp},n) \cap S(d_{src},n)|}{\min(|S(d_{susp},n)|,|S(d_{src},n)|)} \quad (1)$$

where:

- $C_{\_n}$: is the containment of n-gram.
- $S(d_{susp}, n)$: the set of n-grams for the suspicious document (d_{susp}).
- $|S(d_{susp}, n)|$: the number of n-grams in the suspicious document (d_{susp}).
- $S(d_{src}, n)$: the set of n-grams for the source document (d_{src}).
- $|S(d_{src}, n)|$: the number of n-grams in the source document (d_{src}).
- $|S(d_{susp}, n) \cap S(d_{src}, n)|$: the number of shared n-grams between the suspicious (d_{susp}) and source documents (d_{src}).

The containment value is in the range [0,1], where a value of 0 indicates that the pair suspicious and source document have no shared n-grams (no overlap). On the contrary, the value of 1 indicates that the suspicious document is entirely copied from the source document. The containment features can identify cases of copy-pasted and paraphrased texts because they find overlap in word usage between the suspicious document and the source document. Taking the following two sentences, as an example:

- $Sent_A$: "طرح فندق انتركونتيننتال تشرشل لندن مجموعة جديدة من المبادرات",
  London InterContinental Churchill Hotel introduced a new set of initiatives.

– Sent_B: "يقوم فندق انتركونتيننتال تشرشل لندن بتجديد عروضه من خلال طرح مجموعة جديدة من المبادرات".
London InterContinental Churchill Hotel updates its offering with new set of initiatives.

The uni-gram, bi-gram, tri-gram, and 4-gram terms for Sent_A and Sent_B are listed in Table 4, and the containment values C_1, C_2, C_3, and C_4 are listed in Table 5.

Table 4. The 1-gram, 2-gram, and 3-gram terms for Sent_A and Sent_B

n-gram	n-gram terms of Sent_A	n-gram terms of Sent_B
Uni-gram	"طرح", "فندق", "انتركونتيننتال", "تشرشل", "لندن", "مجموعة", "جديدة", "من", "المبادرات"	"يقوم", "فندق", "انتركونتيننتال", "تشرشل", "لندن", "بتجديد", "عروضه", "من", "خلال", "طرح", "مجموعة", "جديدة", "المبادرات"
Bi-gram	"طرح فندق", "فندق انتركونتيننتال", "انتركونتيننتال تشرشل", "تشرشل لندن", "لندن مجموعة", "مجموعة جديدة", "جديدة من", "من المبادرات"	"يقوم فندق", "فندق انتركونتيننتال", "انتركونتيننتال تشرشل", "تشرشل لندن", "لندن بتجديد", "بتجديد عروضه", "عروضه من", "من خلال", "خلال طرح", "طرح مجموعة", "مجموعة جديدة", "جديدة من", "من المبادرات"
Tri-gram	"طرح فندق انتركونتيننتال", "فندق انتركونتيننتال تشرشل", "انتركونتيننتال تشرشل لندن", "تشرشل لندن مجموعة", "لندن مجموعة جديدة", "مجموعة جديدة من", "جديدة من المبادرات"	"يقوم فندق انتركونتيننتال", "فندق انتركونتيننتال تشرشل", "انتركونتيننتال تشرشل لندن", "تشرشل لندن بتجديد", "لندن بتجديد عروضه", "بتجديد عروضه من", "من خلال طرح", "عروضه من خلال", "خلال طرح مجموعة", "طرح مجموعة جديدة", "مجموعة جديدة من", "جديدة من المبادرات"
4-gram	"طرح فندق انتركونتيننتال تشرشل", "فندق انتركونتيننتال تشرشل لندن", "انتركونتيننتال تشرشل لندن مجموعة", "تشرشل لندن مجموعة جديدة", "لندن مجموعة جديدة من", "مجموعة جديدة من المبادرات"	"يقوم فندق انتركونتيننتال تشرشل", "فندق انتركونتيننتال تشرشل لندن", "انتركونتيننتال تشرشل لندن بتجديد", "تشرشل لندن بتجديد عروضه من", "بتجديد عروضه من خلال", "عروضه من خلال طرح", "من خلال طرح مجموعة", "خلال طرح مجموعة جديدة", "طرح مجموعة جديدة من", "مجموعة جديدة من المبادرات"

Table 5. Containment values C_1, C_2, C_3, and C_4 for Sent_A and Sent_B

S(Sent_A, n) ∩ S(Sent_B, n)	|S(Sent_A, n) ∩ S(Sent_B, n)|	|S(Sent_A, n)|	|S(Sent_B, n)|	min (|S(Sent_A, n)|, |S(Sent_B, n)|)	C_n (Sent_A, Sent_B)
"طرح", "فندق", "انتركونتيننتال", "تشرشل", "لندن", "مجموعة", "جديدة", "من", "المبادرات"	9	9	13	9	C_1=9/9=1
"فندق انتركونتيننتال", "انتركونتيننتال تشرشل", "تشرشل لندن", "مجموعة جديدة", "جديدة من", "من المبادرات"	6	8	13	8	C_2=6/8=0.75
"فندق انتركونتيننتال تشرشل", "انتركونتيننتال تشرشل لندن", "تشرشل لندن مجموعة", "مجموعة جديدة من", "جديدة من المبادرات"	4	7	12	7	C_3=4/7=0.57
"فندق انتركونتيننتال تشرشل لندن", "انتركونتيننتال تشرشل لندن مجموعة", "مجموعة جديدة من المبادرات"	2	6	11	6	C_4=2/6=0.33

b. Longest common subsequence feature

The LCS is the length of the longest sequence of words that occurs in both text while preserving the original order. Let d_{susp} denote the suspicious document and d_{src} denote the source document. The LCS corresponds to the longest ordered sequence of shared words between d_{susp} and d_{src}.

In other words, LCS calculates the number of editing operations, including insertions and deletions, needed to transform one text (d_{susp}) into another (d_{src}), where each operation is assigned a unit cost. The LCS value is calculated as shown in (2) [23]:

$$\text{LCS}\,(d_{susp\_i}, d_{src\_j}) = \begin{cases} 0 & \text{,if i = 0,or j = 0} \\ \text{LCS}\,(d_{susp\_i-1}, d_{src\_j-1}) + 1 & \text{if i,j > 0,and dSusp\_i = dSrc\_j} \\ \max(\text{LCS}\,(d_{susp\_i}, d_{src\_j-1}), \text{LCS}\,(d_{susp\_i-1}, d_{src\_j})) & \text{if i,j > 0,and dSusp\_i} \neq \text{dSrc\_j} \end{cases} \quad (2)$$

The LCS value is normalized by dividing it by the word count of the shortest document, either the suspicious document (d_{susp}) or the source document (d_{src}) as defined in (3).

$$LCS_{norm} = \text{LCS}(d_{susp}, d_{src}) \,/\, \min(|d_{susp}|, |d_{src}, n|) \quad (3)$$

For Sent_A, Sent_B, the LCS value is 7, resulting in a normalized LCS value of 7/9=0.78. However, Table 6 provides the process of computing the LCS value for these two sentences.

Table 6. LCS value between Sent_A and Sent_B

	يقوم do	فندق Hotel	انتركونتينينتال Inter-Continental	تشرشل Churchill	لندن Londo n	بتجديد update	عروضه offers	من of	خلال through	طرح Introd -uce	مجموعة set	جديدة new	من of	المبادرات Initiatives
Φ	0	0	0	0	0	0	0	0	0	0	0	0	0	0
طرح introduce	0	0	0	0	0	0	0	0	1	1	1	1	1	1
فندق Hotel	0	1	1	1	1	1	1	1	1	1	1	1	1	1
انتركونتيننتال InterContin ental	0	1	1	1	1	1	1	1	1	1	1	1	1	1
تشرشل Churchill	0	1	1	2	2	2	2	2	2	2	2	2	2	2
لندن London	0	1	1	2	3	3	3	3	3	3	3	3	3	3
مجموعة set	0	1	1	2	3	3	3	3	3	3	4	4	4	4
جديدة new	0	1	1	2	3	3	3	3	3	3	4	5	5	5
من of	0	1	1	2	3	3	3	3	3	3	4	5	6	6
المبادرات Initiatives	0	1	1	2	3	3	3	3	3	3	4	5	6	7

3.3.3. Feature extraction

In this phase, similarity features between a suspicious text and a source text are extracted. Containment features are generated from the training dataset by calculating overlaps among n-grams ranging from unigram, 7-gram overlaps, where each containment feature calculates the number of n-gram phrases between the suspicious and source documents. These features are important to detect copy-past plagiarism and near-copy plagiarism, which has minor modifications to the source text. In addition, the normalized LCS value (LCS_{norm}) is computed.

At the end of this phase, the Primer_Training_Dataset contains 11 features, namely, the suspicious text, source text, plagiarism_class, seven containment features (C_1, C_2, C_3, C_4, C_5, C_6, and C_7) and LCS_{norm}.

3.3.4. Feature selection and correlation analysis

During this step, a subset of informative features is identified by examining correlations among features. The eight features aim to evaluate the similarities between the two texts, consequently indicating a significant correlation. The inclusion of highly correlated could boost the importance of a specific feature.

Features are selected based on pairs with the lowest correlation values. The correlation coefficients range from 0 to 1 and are ranked from low to high. Table 7 shows the correlation matrix of the eight features.

Table 7. Correlation matrix of the features

Feature	C_1	C_2	C_3	C_4	C_5	C_6	C_7	LCS_{norm}
C_1	1.00	0.37	0.36	0.36	0.36	0.36	0.36	0.34
C_2	0.37	1.00	0.99	0.98	0.96	0.94	0.93	0.91
C_3	0.36	0.99	1.00	1.00	0.99	0.97	0.96	0.92
C_4	0.36	0.98	1.00	1.00	1.00	0.99	0.98	0.92
C_5	0.36	0.96	0.99	1.00	1.00	1.00	0.99	0.92
C_6	0.36	0.94	0.97	0.99	1.00	1.00	1.00	0.92
C_7	0.36	0.93	0.96	0.98	0.99	1.00	1.00	0.91
LCS_{norm}	0.34	0.91	0.92	0.92	0.92	0.92	0.91	1.00

From the correlation matrix in Table 7, C_2 and C_7 exhibit the lowest correlations with C_1, together with LCS_{norm}, using a cutoff correlation value of 0.91. At the end of this phase, a new training dataset is constructed using only four similarity features: C_1, C_2, C_7, LCS_{norm}, and plagiarism_class. The resulting dataset is called Final_Training_Dataset. Similarly, the Final_Testing_Dataset is constructed by computing C_1, C_2, C_7, and LCS_{norm} from the Primer_Testing_Dataset.

3.4. Proposed model architectures (support vector machine and artificial neural network)

The suggested approach is evaluated using two classifiers, namely the support vector machine (SVM) classifier and an ANN model. The specifications for these two models are listed in the next subsections.

3.4.1. Baseline model

Plagiarism detection can be formulated as a binary classification problem. In binary classification tasks, a proficient model employed is the SVM. It functions by establishing a hyperplane that separates samples from one class on one side and samples from the opposing class on the other, therefore distinguishing instances of the two classes.

Typically, multiple hyperplanes exist, and the SVM selects the one that most effectively separates the two classes. In binary classification tasks, SVM have demonstrated effective performance [24], [25]. The baseline model is a SVM classifier (SVM_Base Model). The final training dataset, comprising the class label and four similarity features (C_1, C_2, C_7, and LCS_norm), is utilized to train the SVM_Base Model.

3.4.2. Artificial neural network model

Artificial neurons are typically organized in layers, with signals transmitted from the first layer to the final layer. The initial layer corresponds the input layer, the hidden layers are situated in between input layer and output layer.

An ANN model with three layers-an input layer, the last layer servers as output layer, with one or more hidden layer between them-is used in the proposed method. C_1, C_2, C_7, and LCS_norm are the four real-valued features that make up the Final_Training_Dataset. A dense layer of 300 neurons that is entirely linked receives the input features. The output layer then receives these neurons' outputs and uses a sigmoid activation function to predict the class. The model utilizes the Adam optimizer to improve the ANN model's learning process, while binary cross-entropy is employed as the loss function for the plagiarism detection model.

3.4.3. Experimental setup

The open-source Keras (version 3) utilizing TensorFlow as a backend is employed to develop the ANN model, whereas the open-source scikit-learn (version 1.7) Python machine learning library is utilized to construct the SVM-based model. Each experiment is conducted utilizing the GPU accelerator offered by Google Collaboratory. The ANN model is trained over 1000 epochs using batch size of 100. Table 8 summarizes the optimization hyperparameters adopted in the experimental work.

To assess the robustness of the proposed models, each experiment was repeated 10 independent times using different random seeds while keeping all datasets, preprocessing steps, and hyperparameters unchanged. The reported performance corresponds to the mean values across all runs, and 95% confidence intervals (95% CI) were computed for precision, recall, and F1-score.

Table 8. Set of hyper-parameters used during the training step of all the phases

Hyper-parameter	Value
Number of epochs	1000
Loss function	Binary cross-entropy
Optimizer	Adam
Learning rate	0.001
Batch size	100

3.5. Evaluation measures

Model performance is assessed using three metrics: precision, recall, and F1-score. They are computes using the value from the confusion matrix presented in Table 9. The confusion matrix, as shown in Table 8, consists of four values. The two values are associated with labels of the predicted class, whereas the others are associated with labels of the actual class.

Table 9. Confusion matrix

Class	Predicted negative	Predicted positive
Actual negative	True negative (TN)	False positive (FP)
Actual positive	False negative (FN)	True positive (TP)

In this work, two class labels are: “plagiarized” (positive) and “non-plagiarized” (negative). TN is the number of non-plagiarized samples predicted correctly as non-plagiarized, and TP is the number of plagiarized samples predicted correctly as plagiarized. FN is the number of plagiarized samples incorrectly classified as non-plagiarized, whereas FP is the number of non-plagiarized samples incorrectly classified as plagiarized. Table 10 lists the formulas used to compute the performance metrics.

Table 10. Performance measures

Performance measure	Definition
Recall (R)	$Recall = TP \times \frac{100}{TP+FN}$
Precision (P)	$Precision = TP \times \frac{100}{TP+FP}$
F1-score	$F1 - score = 2 \times \frac{P \times R}{P+R}$

4. RESULT AND DISCUSSION

The SVM and ANN models were evaluated using the Final_Testing_Dataset, and the mean precision, recall, and F1-score obtained across the repeated experiments are summarized in Table 11 together with their corresponding 95% confidence intervals.

Table 11. Comparison of SVM and ANNs models with existing models

Model	Precision (%)	Recall (%)	F1-score (%)
Proposed SVM (base model)	91.1 (95% CI: 90.8–91.4)	100.0 (95% CI: 100.0–100.0)	95.0 (95% CI: 94.8–95.2)
Proposed ANNs	92.2 (95% CI: 91.9–92.5)	100.0 (95% CI: 100.0–100.0)	96.1 (95% CI: 95.9–96.3)
Magooda_2 [26]	85	83	84
Magooda_3 [26]	85	76	80
Magooda_1 [26]	81	79	80
Baseline_competition [10]	99	54	69
Alzahrani [27]	83	53	65
Palkovskii_1 [28]	100	54	70
Palkovskii_3 [28]	66	59	62
Palkovskii_2 [28]	56	59	58

Table 11 compares the results of the SVM and ANN models with other existing models, including three methods presented by Magooda *et al*. [26], one method by Alzahrani [27], three methods by Palkovskii and Belov [28], and the baseline competition results reported in Bensalem *et al*. [10]. The three basic modules of Magooda *et al*. [26] are Magooda_1 which is based on retrieving candidate source documents, while Magooda_2 is an alignment module and Magooda_3 is a filtering module. The three methods of Palkovskii used clustering with different modules; Palkovskii_1 is a fingerprinting-based approach, while Palkovskii_2 combined fingerprinting and sliding-window, and Palkovskii_3 used a multi-type n-gram module with adaptive clustering.

Only the three techniques proposed by Magooda *et al*. [26] can identify instances of obfuscation involving word shuffling. By contrast, the methods proposed by Palkovskii and Belov [28] do not explicitly address this type of obfuscation, which explains their low recall values. For cases without obfuscation, all approaches achieve high recall rates, but obfuscation through paraphrasing is the most difficult to identify.

The findings indicate that the proposed SVM and ANN models outperform other methods in terms of F1-score. In addition, the ANN model performs better than the SVM (base model) in terms of F1-score. The former provides an improvement of approximately 12% compared with the best result among the other models, as obtained by the Magooda _2 method. This enhancement indicates that combining containment features with the LCS feature improves the effectiveness of detecting plagiarism in Arabic texts.

LCS and n-gram overlap features exhibit advantages and disadvantages in plagiarism detection. n-gram features are simple and scalable, but they lack context and are less effective in identifying paraphrasing, especially for lower-order n-grams. Although the suggested method shows that higher-order n-grams work effectively for local word sequences, they suffer from data sparsity when applied to Arabic text.

The effectiveness of the introduced method lies in its ability to capture the co-occurrence of adjacent words and recognize that specific word combinations in Arabic convey distinct meanings beyond individual tokens. By utilizing n-gram overlap and LCS features within an ANN framework, the model maintains structural context and handles the complex morphology of Arabic better than simple bag-of-words baselines. Furthermore, the use of neural networks helps mitigate specific algorithmic weaknesses. However, trade-offs remain. Employing high-order n-grams can be computationally demanding. Additionally, although the lexical approach is highly effective for structural matching, it does not fully address semantic plagiarism, such as deep paraphrasing using synonyms. Future testing is required to ensure these the generalizability of the results across different domains and Arabic dialectal variations.

5. ERROR ANALYSIS

The results show that the model performance is strongly affected by the type of plagiarism being considered in the experimented document. When considering different categories, plagiarism using paraphrasing was the most difficult to detect. This is mainly because paraphrased text changes vocabulary and sentence structure while the original meaning is kept intact, which makes a challenge for approaches that depend heavily on surface-level of text similarity. Word shuffling is less challenging. Although the order of words is altered, the lexical content itself remains unchanged, making it easier for models that rely on word-level matching to identify similarities. Approaches such as those introduced by Magooda *et al*. [26] perform reasonably well in this context, though their effectiveness decreases when deeper semantic changes are involved.

The least challenging cases are those with no obfuscation, where the copied content is nearly identical to the source. In these situations, most models achieve high recall, as the overlap between texts is straightforward to detect.

The SVM and the proposed ANN model achieve a recall of 100%, suggesting that they successfully identify all plagiarism instances, including those involving obfuscation. This performance specifies that the models most likely capturing more informative and representative features compared to simple word overlap, enabling them to better handle both structural and semantic variations.

However, the methods proposed by Palkovskii and Belov [28] as reported in Bensalem *et al.* [10] show mainly lower recall values, approximately down to 54%, which reflects a higher number of missed cases. This limitation can be attributed to their weaker ability to recognize obfuscated plagiarism, particularly when paraphrasing is involved. These results confirm that paraphrasing remains the key challenges of plagiarism variant requiring detection, highlighting the importance of incorporating stronger semantic analysis techniques in future work.

6. CONCLUSION

Plagiarism is a pervasive problem that exists in different languages. Using the internet and technological development tools makes it easier for several users to use texts defined by others without referring to the original source and then commit plagiarism. The proposed model for plagiarism detection combines different lexical features, including n-grams, containments, and LCS with ANNs to detect plagiarism in Arabic texts at the document level. Our framework can be seamlessly incorporated into existing plagiarism detection tools to enhance accuracy for Arabic texts. The experimental findings indicates that the introduced model archives high performance in detecting plagiarism cases with a precision value of 92% for the ANNs model compared to the state-of-the-art approach. As part of future work, we suggest improving the proposed model to detect various types of plagiarism along with combining sematic features with lexical features to enhance the overall capability of plagiarism detection and reducing errors. Furthermore, applying the proposed model for cross-lingual plagiarism detection and integration with LLMs.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Marwah Alian	✓	✓			✓	✓	✓		✓	✓		✓	✓	
Dana Halabi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Hadeel Rida Alshboul						✓	✓			✓	✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created in this study.

REFERENCES

[1] Y. A. Abdelrahman, A. Khalid, and I. M. Osman, "A Method for Arabic Documents Plagiarism Detection," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 15, no. 2, p. 79, 2017.

[2] E. Al-Thwaib, B. H. Hammo, and S. Yagi, "An academic Arabic corpus for plagiarism detection: design, construction and experimentation," *International Journal of Educational Technology in Higher Education*, vol. 17, no. 1, p. 1, Dec. 2020, doi: 10.1186/s41239-019-0174-x.

[3] M. J. Al-Khazaleh, M. Alian, and M. A. Jaradat, "Sentiment analysis of imbalanced Arabic data using sampling techniques and classification algorithms," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 607–618, Feb. 2024, doi: 10.11591/eei.v13i1.5886.

[4] M. M. Biltawi, S. Tedmori, and A. Awajan, "Arabic Question Answering Systems: Gap Analysis," *IEEE Access*, vol. 9, pp. 63876-63904, 2021, doi: 10.1109/ACCESS.2021.3074950.

[5] E. M. B. Nagoudi, A. Khorsi, H. Cherroun, and D. Schwab, "2L-APD: A two-level plagiarism detection system for Arabic documents," *Cybernetics and Information Technologies*, vol. 18, no. 1, pp. 124–138, Mar. 2018, doi: 10.2478/cait-2018-0011.

[6] A. M. E. T. Ali, H. M. D. Abdulla, and V. Snasel, "Survey of Plagiarism Detection Methods," in *2011 Fifth Asia Modelling Symposium*, Manila, Philippines, May 2011, pp. 39–42, doi: 10.1109/AMS.2011.19.

[7] A. Amirzhanov, C. Turan, and A. Makhmutova, "Plagiarism types and detection methods: a systematic survey of algorithms in text analysis," *Frontiers in Computer Science*, vol. 7, pp. 1-20, 2025, doi: 10.3389/fcomp.2025.1504725.

[8] A. S. Bin-Habtoor and M. A. Zaher, "A Survey on Plagiarism Detection Systems," *International Journal of Computer Theory and Engineering*, pp. 185–188, 2012, doi: 10.7763/ijcte.2012.v4.447.

[9] A. Taloni, V. Scorcia, and G. Giannaccare, "Modern threats in academia: evaluating plagiarism and artificial intelligence detection scores of ChatGPT," *Eye (Basingstoke)*, vol. 38, no. 2, pp. 397–400, Feb. 2024, doi: 10.1038/s41433-023-02678-7.

[10] I. Bensalem, I. Boukhalfa, P. Rosso, L. Abouenour, K. Darwish, and S. Chikhi, "Overview of the AraPlagDet PAN@FIRE2015 shared task on Arabic plagiarism detection," in *CEUR Workshop Proceedings*, 2015, vol. 1587, pp. 111–122.

[11] I. H. Khan, M. A. Siddiqui, K. M. Jambi, and A. A. Bagais, "A Framework for Plagiarism Detection in Arabic Documents," in *Proceedings of the Conference on Computer Science & Information Technology*, pp. 01–09, Jan. 2015, doi: 10.5121/csit.2015.50201.

[12] C. Liu, C. Chen, J. Han, and P. S. Yu, "GPLAG: detection of software plagiarism by program dependence graph analysis" in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2006, pp. 872–881, doi: 10.1145/1150402.1150522.

[13] E. Hattab, "Cross-Language Plagiarism Detection Method: Arabic vs. English," in *Proceedings - 2015 International Conference on Developments in eSystems Engineering (DeSE)*, Dec. 2016, pp. 141–144, doi: 10.1109/DeSE.2015.25.

[14] M. A. Siddiqui, I. H. Khan, K. M. Jambi, S. O. Elhaj, and A. Bagais, "Developing an Arabic Plagiarism Detection Corpus," in *Proceedings of the Conference on Computer Science & Information Technology*, Dec. 2014, pp. 261–269, doi: 10.5121/csit.2014.41221.

[15] A. A. A. Yousef and M. J. A. Aziz, "Enhanced Tf-Idf Weighting Scheme For Plagiarism Detection Model for Arabic Language," *Australian Journal of Basic and Applied Sciences*, vol. 9, pp. 90–96, 2015.

[16] B. Ghanem, L. Arafeh, P. Rosso, and F. Sánchez-Vega, "HYPLAG: Hybrid arabic text plagiarism detection system," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10859, pp. 315–323, 2018, doi: 10.1007/978-3-319-91947-8_33.

[17] S. Alzahrani and N. Salim, "Fuzzy semantic-based string similarity for extrinsic plagiarism detection: Lab report for PAN at CLEF 2010," in *CEUR Workshop Proceedings*, vol. 1176, 2010.

[18] D. Suleiman, A. Awajan, and N. Al-Madi, "Deep learning based technique for plagiarism detection in Arabic texts," in *Proceedings - 2017 International Conference on New Trends in Computing Sciences, ICTCS 2017*, vol. 2018, pp. 216–222, 2017, doi: 10.1109/ICTCS.2017.42.

[19] E. H. Omoush, A. Shdefat, N. Al-Madi, and B. Hammo, "Arabic Paraphrasing Detection Using Multiple Extracted Features," in *2023 14th International Conference on Information and Communication Systems, ICICS 2023*, Nov. 2023, pp. 01–06, doi: 10.1109/ICICS60529.2023.10330486.

[20] M. A. El-Rashidy, R. G. Mohamed, N. A. El-Fishawy, and M. A. Shouman, "An effective text plagiarism detection system based on feature selection and SVM techniques," *Multimedia Tools and Applications*, vol. 83, no. 1, pp. 2609–2646, 2024, doi: 10.1007/s11042-023-15703-4.

[21] H. Alshammari, A. El-Sayed, and K. Elleithy, "AI-Generated Text Detector for Arabic Language Using Encoder-Based Transformer Architecture," *Big Data and Cognitive Computing*, vol. 8, no. 3, p. 32, Mar. 2024, doi: 10.3390/bdcc8030032.

[22] I. Zeroua, M. Boudchiche, A. Mazroui, and A. Lakhouaja, "Improving Arabic light stemming algorithm using linguistic resources," *The Second National Doctoral Symposium on Arabic Language Engineering (JDILA'2015)*, Fez, Morocco, 2015.

[23] P. Clough and M. Stevenson, "Developing a corpus of plagiarised short answers," *Language Resources and Evaluation*, vol. 45, no. 1, pp. 5–24, Mar. 2011, doi: 10.1007/s10579-009-9112-1.

[24] W. Alabbas, H. M. Al-Khateeb, A. Mansour, G. Epiphaniou, and I. Frommholz, "Classification of colloquial Arabic tweets in real-time to detect high-risk floods," in *2017 International Conference on Social Media, Wearable and Web Analytics (Social Media)*, Jun. 2017, pp. 1–8, doi: 10.1109/SOCIALMEDIA.2017.8057358.

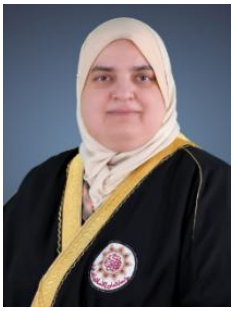
[25] A. Al-Hassan and H. Al-Dossari, "Detection of hate speech in Arabic tweets using deep learning," *Multimedia Systems*, vol. 28, no. 6, pp. 1963–1974, Dec. 2022, doi: 10.1007/s00530-020-00742-w.

[26] A. Magooda, A. Y. Mahgoub, M. Rashwan, M. B. Fayek, and H. Raafat, "RDI system for extrinsic plagiarism detection (RDI-RED): Working notes for PAN-AraPlagDet at FIRE 2015," in *CEUR Workshop Proceedings*, vol. 1587, pp. 126–128, 2015.

[27] S. Alzahrani, "Arabic plagiarism detection using word correlation in N-grams with K-overlapping approach," in *CEUR Workshop Proceedings*, vol. 1587, pp. 123–125, 2015.
[28] Y. Palkovskii and A. Belov, "Developing High-Resolution Universal Multi-Type N-Gram Plagiarism Detector," in *Working Notes Papers of the CLEF 2014 Evaluation Labs*, CEUR Workshop Proceedings, vol. 1180, pp. 984–989, 2014.

BIOGRAPHIES OF AUTHORS

Marwah Alian is a faculty member at the Department of Computer Science, Faculty of Information Technology, the World Islamic Sciences and Education University, Amman, Jordan. She received her B.Sc. in Computer Science from the Hashemite University in 1999. Then, she received M.Sc. degree in Computer Science from the University of Jordan in 2007. In 2021, she received her Ph.D. from Princess Sumaya University for Technology. She has a number of publications in e-learning systems, data mining, mobile applications, adaptive learning, mobile learning, and natural language processing. She can be contacted at email: marwah.alian@wise.edu.jo and marwah2001@yahoo.com.



Dana Halabi assistant professor with a Ph.D. in Computer Science and over 10 years of combined industrial and academic experience. Proven expertise in data science, machine learning, deep learning, natural language processing, computer vision, generative AI models, and RAG implementation. Her research focuses on advanced applications of deep learning in both Arabic natural language processing and the medical domain, evidenced by work on Arabic LLMs, text-to-SQL, and ongoing research in skin cancer detection and brain tumor segmentation. She can be contacted at email: dana.halabi@wise.edu.jo.

Hadeel Rida Alshboul is a full-time Lecturer at Department of Computer Science, Faculty of Information Technology, The World Islamic Sciences and Education University (Jordan). She received her B.Sc. degree in Computer Science from AL Balqa Applied University in 2020, and then she received her M.Sc. degree in Computer Science from Princess Sumaya University for Technology in 2022. Her research interests include machine learning, artificial intelligence, and e-learning systems. She can be contacted at email: hadeel.alshboul@wise.edu.jo.